1. git 是什么？
   1. 是目前世界上最先进的版本分布式版本控制系统（没有之一）
2. Git怎么安装？
   1. 1.账号注册网址： <http://qithub.com>
   2. 配置系统
      1. Linux sudo apt-get install git
      2. Mac os 就是直接从AppStore安装Xcode，Xcode集成了Git，不过默认没有安装，你需要运行Xcode，选择菜单“Xcode”->“Preferences”，在弹出窗口中找到“Downloads”，选择“Command Line Tools”，点“Install”就可以完成安装了
      3. Windons [应用网盘](https://pan.baidu.com/s/1kU5OCOB?errno=0&errmsg=Auth%20Login%20Sucess&&bduss=&ssnerror=0&traceid=#list/path=%2Fpub%2Fgit) 安装完成后，在开始菜单里找到“Git”->“Git Bash”，蹦出一个类似命令行窗口的东西，就说明Git安装成功！
         1. 装完成后，还需要最后一步设置，在命令行输入：
         2. $ git config --global user.name "Your Name"
         3. $ git config --global user.email "email@example.com"
3. Git的使用步骤？

网易邮箱的 账号 [13383833163@163.com](mailto:13383833163@163.com) zhanglei19951127

Github 账号 zhangjinlei1127 zhanglei19951127

1.设置：

1.配置用户名 git config --global user.name “zhangjinlei1127”

2.配置邮箱 git config –global user.email “13383833163@163.com” (配置一次就可以了)

2.版本库 **本地操作**

1.什么是版本库：

又名仓库，可以理解为一个目录，这个目录里面的所有文件都可以被git管理起来，每个文件的修改，删除，Git都可以跟踪，以便任何时刻都有一个追踪的历史，或者将来某一个时刻还原

2.创建版本库 在一个合适的地方，创建一个空目录 cd创建一个目录 使用 git init 把一个目录可以变成一个可有git管理的仓库 注意：一定不要手动改变.git目录里面的文件时

MAIBENBEN@DESKTOP-1P7R2QS MINGW64 ~/Desktop/测试git

$ git init

Initialized empty Git repository in C:/Users/MAIBENBEN/Desktop/测试git目录/.git/

1. 把文件添加到版本库
   1. 在仓库目录里创建一个git.docx 文件 添加上相应的内容
   2. 把文件添加到仓库 git add git .docx
   3. 把文件提交到仓库 git commit –m “注释信息—一般为修改的代码的功能”
2. git 版本命令
   1. git status 查看当仓库的一个状态
   2. git diff 查看修改的内容（不常用）
   3. git log 可以查看显示从最近道德最远的提交日志
      1. 显示 作者 邮箱 提交的内容的编号 提交内容的时间 内容注释
         1. git log –-pretty=oneline 只显示提交的编号 内容注释
   4. \*\*\*\*\*重要的额内容\*\*\*\*
      1. 版本回退
         1. git reset --hard HEAD^ 回到上一个版本
         2. git reset –hard HEAD^^ 回退上上版本
         3. 回退上100个版本 git reset –hard HEAD-100
         4. 回到具体的版本 git reset –hard 具体的版本号
         5. Git reflog 记录每一次命令
      2. Git checekout – filename （有些情况不需要退还版本 可以直接撤回操作）
         1. 如果文件字修改后还没有放到缓存区，现在，可以撤销修改就会回到和版本库·一模一样的状态
         2. 如果文件添加到暂存区后，又做了修改，现在，撤销就回到添加到暂存区后的状态
         3. 总之，就是让这个filename文件回到最后git commit 或者 git add 的状态
      3. 工作区 版本区 缓冲区

版本库

commit

工作区

Git.docx 文档

缓存区

Git.docx 文档

Master 分支

Add

**远程仓库**

* + - 1. 创建SSH KEY 是黑屏终端输入 ssh-keygen –t rsa –C “自己的GitHub的注册邮箱”
      2. 记录.shh的目录位置 /c/Users/MAIBENBEN/.ssh
      3. 输入GitHub注册的密码（注意：密文输入，黑屏终端不会显示密码）
      4. .ssh
         1. Id\_rsa 私钥
         2. Id\_rsa.pub 公钥
         3. ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDq5f41/p6Fjvsdd3sgQBGMTgMIBSOPs1Ti+SRw0ahj/sFI2/qG2s/XM/KhhDBwnNaai2qcslWGgi4IDE2CWwRe8mN0kqCp58CXFw9wXGyRvZcV1b2vZeVrabdwzTqT6SWRXJumMBXK0XiUDqJcco2ikDTkK8i8tUwMdtB67Ai3Kw7SAb1l8hQms593dvlkN5q7gd5cAsmhsm9ANzioWBRze7/udQ9Aa4EOhqCYSTLHx7rGVpj6e0D/p1vtCbcnwzzKzgJz3BeMZLqpt+C1YGgWtO5dOKq3AMenmLX1fycaeAQPvYklho1Lq2oWB4jurg2HxIGMVVKOVWHuCJPiPomj 13383833163@163.com
      5. 登录网页GitHub 进行设置 ssh 设置
      6. 测试钥匙是否可以使用 ssh –T git@github.com

1. 创建远程仓库 在网页上常见仓库的名字
2. 广联远程仓库 git remote add origin [git@github.com:zhangjinlei1127/git.git](mailto:远程仓库地址git@github.com:zhangjinlei1127/git.git) closeor download > use ssh远程仓库地址（关联的是自己的电脑 ）
3. 删除关联 git remote rm origin
4. 推送本地库到远程仓库 git push origin master
   1. 注意：需要先把远程库拉到本地，否则会报错
   2. 忽略特殊文件件 .gitignore 里面填写大文件包的名字 节约上传的时间 （固定文件名）
5. 先拉取远程内容到本地库 git pull origin master （老了不好用了）
   1. fatal: refusing to merge unrelated histories 报错
   2. **git pull origin master --allow-unrelated-histories**

先pull，因为两个仓库不同，发现refusing to merge unrelated histories，无法pull

因为他们是两个不同的项目，要把两个不同的项目合并，git需要添加一句代码，在git pull，

这句代码是在git 2.9.2版本发生的，最新的版本需要添加--allow-unrelated-histories

git pull origin master --allow-unrelated-histories

1. git clone [git@github.com:zhangjinlei1127/git.git](mailto:git@github.com:zhangjinlei1127/git.git) 克隆
2. 从零开发‘
   1. 先有远程库
   2. 从远程库克隆 git clone 远程库的地址 git clone [git@github.com:zhangjinlei1127/git.git](mailto:git@github.com:zhangjinlei1127/git.git)

**分支管理**

作用：

代码：

1. 创建一个分支 git branch 分支名
2. 切换分支 git checkout 分支名
3. 创建与切换一起运行 git checkout –b 分支名
4. 查看当前分支 git branch
   1. 会列出当前分支的名称
   2. 标星号为当前的分支
5. 切换分zhanglei 修改git 并提交（添加的内容）
6. 切换主轴master 分支 git checkout master
   1. 添加的内容不存在
   2. 因为是在分支zhanglei操作的
7. 将分支的东西合并到master分支上 git merge dev

**标签管理**